گزارش فاز دوم پروژه درس یادگیری ماشین

تهیه کننده: امیرحسین کارگران خوزانی

شماره دانشجویی: ۹۹۲۰۱۱۱۹

تیر ماه ۱۴۰۰

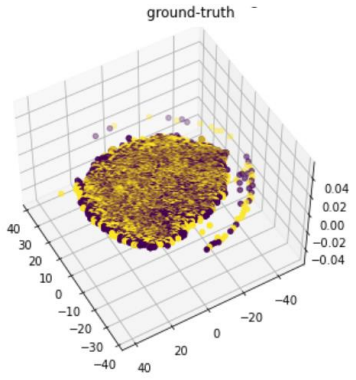
فاز دوم

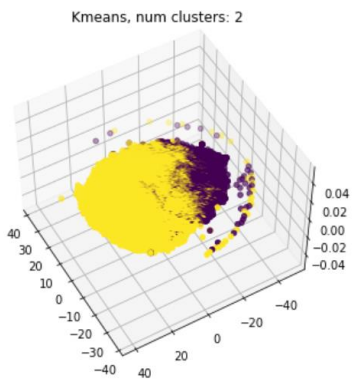
- در پارت اول مدل‌های یادگیری ماشین بدون نظارت را بررسی می‌کنیم.
- در پارت دوم تاثیر دادگان بیشتر و عملیات fine-tune بر روی مدل MLP را بررسی می‌کنیم.

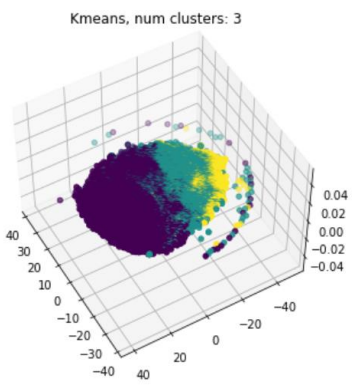
پارت اول: الگوریتم‌های دسته بندی

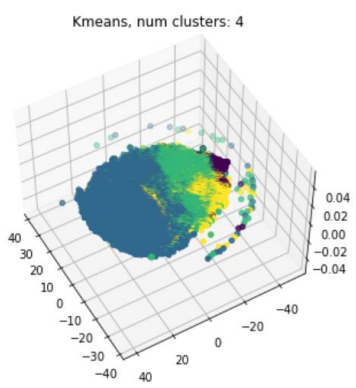
- ۳ الگوریتم Kmeans، GMM و KmeansMiniBatch (در کوئرا اجازه داده شد) برای بررسی انتخاب شدند و برای هر یک به ازای ۲، ۳ و ۴ کلاستر و برای هر یک از روش‌های BOW و W2V اجرا می‌شود.
- با استفاده از الگوریتم TSNE (طبق تجربه قبلی نسبت به PCA شهود بهتری می‌دهد) به ۲ بعد کاهش انجام شده است و این دادگان درست است که در ادامه درqube نمایش داده شده اما در ۲ بعد هستند.
- برای حالت دو کلاسه چند متریک مختلف که به نقل از sklearn برای دسته بندی مناسب هستند انتخاب شد و مقایسه‌ای بین آن‌ها در ادامه انجام می‌شود (منبع)

Kmeans: BOW

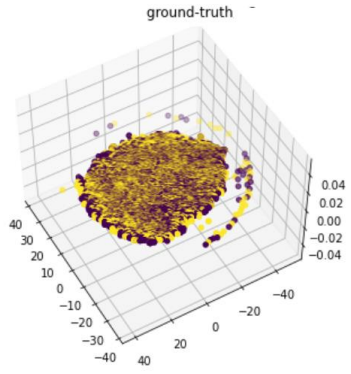

ground-truth

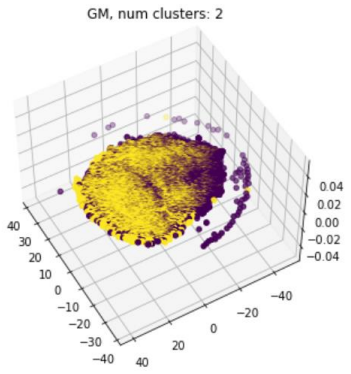

Kmeans, num clusters: 2

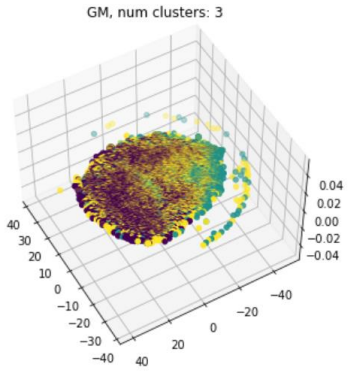

Kmeans, num clusters: 3

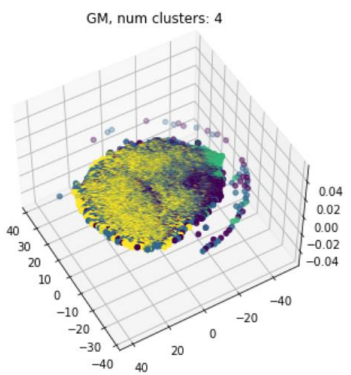

Kmeans, num clusters: 4

GMM: BOW

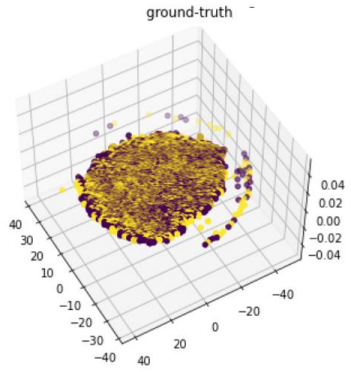

ground-truth

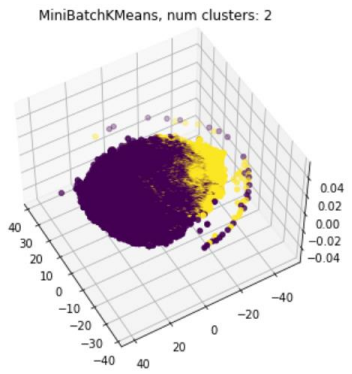

GM, num clusters: 2

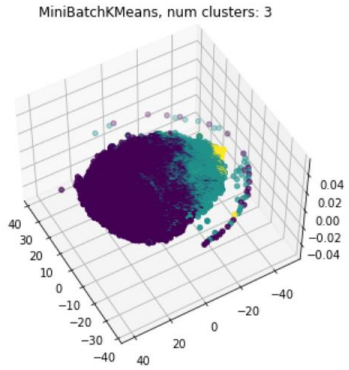

GM, num clusters: 3

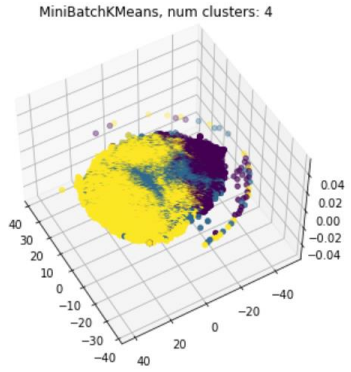

GM, num clusters: 4

KmeansMiniBatch : BOW

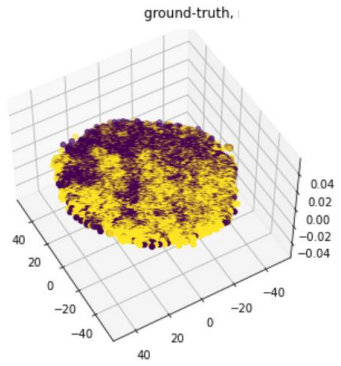


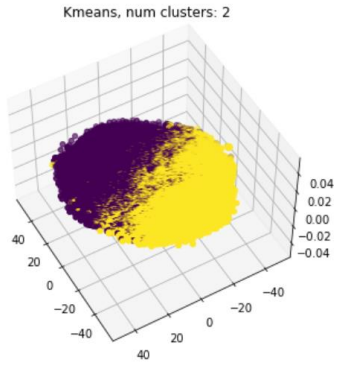


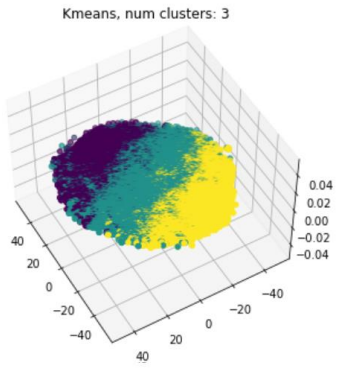


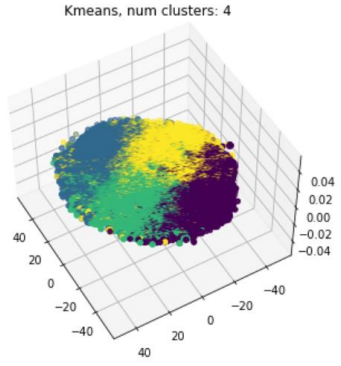


Kmeans: W2V

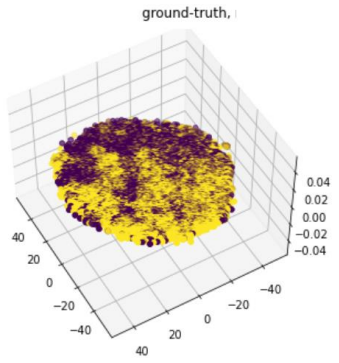

ground-truth,
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

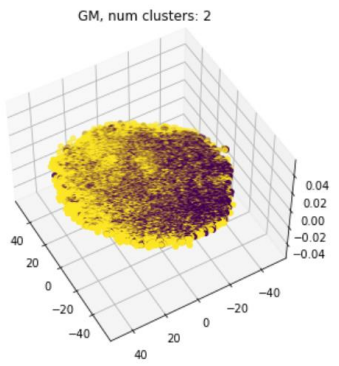

Kmeans, num clusters: 2
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

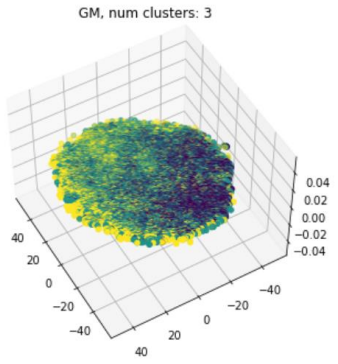

Kmeans, num clusters: 3
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

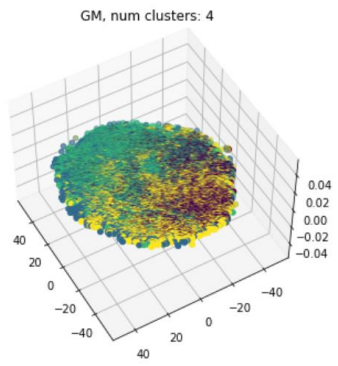

Kmeans, num clusters: 4
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

GMM: W2V

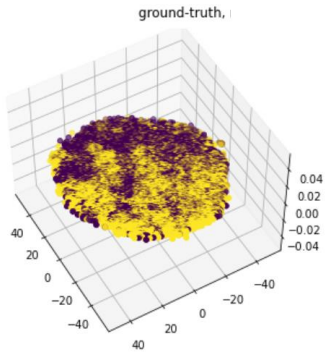

ground-truth,
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

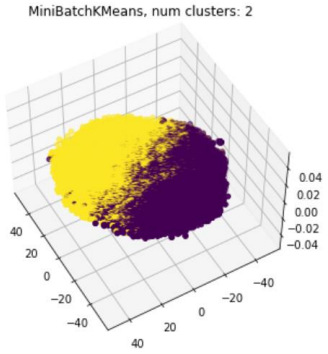

GM, num clusters: 2
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

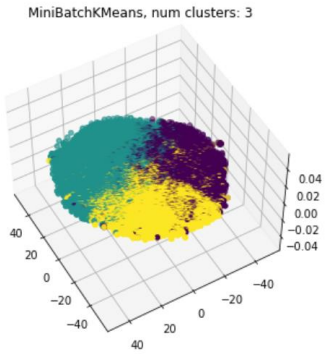

GM, num clusters: 3
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

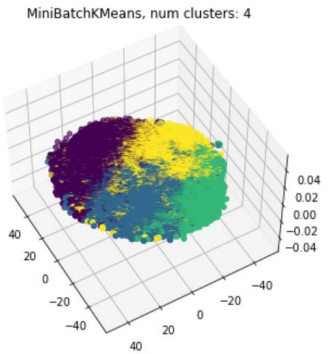

GM, num clusters: 4
0.04
0.02
0.00
−0.02
−0.04
40
20
0
−20
−40
40
20
0
−20
−40

KmeansMiniBatch : W2V









آنالیز دسته بند دو کلاسه

- به این منظور تابع analysis را نوشتیم که معیارهای بیان شده در این لینک را محاسبه می‌کند. از انجا که ممکن است کلاس ۰ و ۱ جا به جا شوند و برعکس مچ شده باشند ما یکبار بین برچسب‌های پیش‌بینی شده و اصل برچسب‌ها نتیجه‌ها را محاسبه کردیم و یک بار هم یکی از کلاس‌ها را برعکس کردیم و ماکسیمم را گزارش کردیم.

```
def analysis(labels, predictions):
    predictions = np.array(predictions)
    predictions2 = ~ np.array(predictions) + 2

    print("Accuracy: \n", max(accuracy_score(labels, predictions), accuracy_score(labels, predictions2)))
    print("Rand Score: \n", max(rand_score(labels, predictions), rand_score(labels, predictions2)))
    print("Adjusted Rand Score: \n", max(adjusted_rand_score(labels, predictions), adjusted_rand_score(labels, predi
    print("Adjusted Mutual Info: \n", max(adjusted_mutual_info_score(labels, predictions), adjusted_mutual_info_scor
    print("Homogeneity Score: \n", max(homogeneity_score(labels, predictions), homogeneity_score(labels, predictions
    print("Completeness Score: \n", max(completeness_score(labels, predictions), completeness_score(labels, predicti
    print("V Measure Score Score: \n", max(v_measure_score(labels, predictions), v_measure_score(labels, predictions
```

ادامه آنالیز دسته بند دو کلاسه

- Accuracy که همان تعداد تشخیص‌های صحیح به کل تشخیص‌هاست.
- Rand index و Adjust شده آن نشان می‌دهد چقدر کلاس‌های ما به کلاس‌های اصلی شبیه هستند و این کار را با شماردن زوج‌ها انجام می‌دهد.

If C is a ground truth class assignment and K the clustering, let us define a and b as:

- a, the number of pairs of elements that are in the same set in C and in the same set in K
- b, the number of pairs of elements that are in different sets in C and in different sets in K

The unadjusted Rand index is then given by:

$$\mathrm{RI} = \frac{a+b}{C_2^{n_{samples}}} \qquad \mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]}$$

- Adjusted Mutual info Score نشان می‌دهد به چه اندازه فرکانس‌های مشاهده شده با کلاس‌های اصلی تفاوت دارد.

$$\mathrm{MI}(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|}\frac{|U_i \cap V_j|}{N}\log\left(\frac{N|U_i \cap V_j|}{|U_i||V_j|}\right) \qquad \mathrm{NMI}(U,V) = \frac{\mathrm{MI}(U,V)}{\mathrm{mean}(H(U),H(V))}$$

خروجی تابع analysis برای BOW

Kmeans

```
Accuracy:
 0.502844444444444
Rand Score:
 0.5000050707299668
Adjusted Rand Score:
 1.7462201752378123e-05
Adjusted Mutual Info:
 2.1148680399360068e-05
Homogeneity Score:
 3.450574636628788e-05
Completeness Score:
 4.618141885943255e-05
V Measure Score Score:
 3.9498830366416686e-05
```

GMM

```
Accuracy:
 0.512
Rand Score:
 0.500276895042112
Adjusted Rand Score:
 0.0005561795579130826
Adjusted Mutual Info:
 0.0004687535634002723
Homogeneity Score:
 0.0004662415380707 4074
Completeness Score:
 0.0005062793166172782
V Measure Score Score:
 0.0004854362683024762
```

KmeansMiniBatch

```
Accuracy:
 0.5008888888888889
Rand Score:
 0.4999904689240008
Adjusted Rand Score:
 -9.766296226834143e-06
Adjusted Mutual Info:
 -1.4632920557867303e-05
Homogeneity Score:
 3.793844150630643e-06
Completeness Score:
 5.640713311406485e-06
V Measure Score Score:
 4.536511073884152e-06
```

خروجی تابع analysis برای W2V

Kmeans

```
Accuracy:
 0.6012444444444445
Rand Score:
 0.5204902192888237
Adjusted Rand Score:
 0.040980481130269905
Adjusted Mutual Info:
 0.02985000012982743
Homogeneity Score:
 0.029843807973999097
Completeness Score:
 0.029887350643074843
V Measure Score Score:
 0.0298655643772453
```

GMM

```
Accuracy:
 0.5041333333333333
Rand Score:
 0.5000230582901842
Adjusted Rand Score:
 4.611851050913743e-05
Adjusted Mutual Info:
 3.32781273056165e-05
Homogeneity Score:
 4.930676388059054e-05
Completeness Score:
 4.9310644096926 6e-05
V Measure Score Score:
 4.930870391242276e-05
```

KmeansMiniBatch

```
Accuracy:
 0.6035777777777778
Rand Score:
 0.5214460775671558
Adjusted Rand Score:
 0.04289216072808916 4
Adjusted Mutual Info:
 0.03117745417205316 4
Homogeneity Score:
 0.03118977730046413 5
Completeness Score:
 0.03119619627991645 4
V Measure Score Score:
 0.03119298645996144
```

نتیجه گیری

- در BOW الگوریتم GMM بهتر عملکرده است.
- در W2V الگوریتم Kmeans بهتر عملکرده است.
- الگوریتم ها به نقاط ابتدایی وابسته هستند و الگوریتم KmeansMiniBatch نسبت به پارامتر batch نیز حساس است.

نتیجه گیری بین دو کلاستر و سه کلاستر

- در هر ۳ الگوریتم با بررسی چند قمت مختلف به صورت دستی مشاهده شد که کلاستر سوم دسته کوچکتری نسبت به دو کلاستر بزرگتر خواهد بود.
- ابتدا فکر کردم که معمولا نظراتی در آن دسته با برچسب سوم قرار می‌گیرد که ممکن است هم منفی و یا مثبت باشد. اما مشاهده کردم که به نظر این بایاس من است که دلم میخواهد این نظرات چنین نظراتی باشند و مشاهده علمی‌ای نیست. اما می‌توان گفت که دسته سوم دسته کوچکتری است و دسته‌های اول و دوم با دسته‌های اول و دوم در کلاستر دوتایی اشتراک زیادی دارند.

پارت دوم: Fine-tuning

- ما ابتدا با استفاده از WordtoVec تنها بر روی همین دادگان محدود مدل MLP خود را با hyper parameter tuning آموزش می‌دهیم و بهترین دقت روی دادگان validation را گزارش می‌کنیم.

MLP: W2V

```
parameter_space = {
    'hidden_layer_sizes': [(50,50,50), (50,100,50),(10,30,10),(20,), (50,), (100,), (150,)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05, 0.1],
    'learning_rate': ['constant','adaptive'],
}

best_param = {
    'hidden_layer_sizes': parameter_space['hidden_layer_sizes'][-1],
    'activation': parameter_space['activation'][-1],
    'solver': parameter_space['solver'][-1],
    'alpha': parameter_space['alpha'][-1],
    'learning_rate': parameter_space['learning_rate'][-1],
}

best_score = 0
```

```
for hls in parameter_space['hidden_layer_sizes']:
    for ac in parameter_space['activation']:
        for so in parameter_space['solver']:
            for al in parameter_space['alpha']:
                for lr in parameter_space['learning_rate']:
                    clf = MLPClassifier(hidden_layer_sizes=hls,learning_rate=lr,alpha=al,solver=so,activation=ac,ma
                    clf.fit(w2v_sentences_train2, Y_train2)
                    score = accuracy_score(Y_val2, clf.predict(w2v_sentences_val2))

                    if score > best_score:
                        best_score = score
                        best_param['hidden_layer_sizes'] = hls
                        best_param['activation'] = ac
                        best_param['solver'] = so
                        best_param['alpha'] = al
                        best_param['learning_rate'] = lr
print(best_score)
print(best_param)
```

Fine-tuning ادامه

- بهترین دقت روی دادگان validation برابر ۰٫۶ و بر روی دادگان تست برابر ۰٫۴ است.

```
0.6
{'hidden_layer_sizes': (20,), 'activation': 'relu', 'solver': 'adam', 'alpha': 0.1, 'learning_rate': 'adaptive'}
```

```
1 analysis(Y_val2, clf.predict(w2v_sentences_val2))
```

```
Report Classification:
              precision    recall  f1-score   support

           0       0.44      0.72      0.55        25
           1       0.22      0.08      0.12        25

    accuracy                           0.40        50
   macro avg       0.33      0.40      0.33        50
weighted avg       0.33      0.40      0.33        50

Matrix Confusion:
 [[18  7]
 [23  2]]
Accuracy:
 0.4
```

ادامه Fine-tuning

- سپس بردارها را با استفاده از Googlenews-Wordtovec به فضای ۳۰۰ بعدی می‌بریم که مناسب داده شدن به ورودی بهترین مدل MLP فاز اول باشد.
- با استفاده از تابع partial_fit دادگان این مدل را نیز به بهترین مدل می‌دهیم و این فرآیند را چند بار انجام می‌دهیم.
- در نهایت دقت بر دادگان تست برابر ۰٫۸۴ خواهد شد.

```
1 analysis(Y_val2, loaded_model.predict(w2v_sentences_val2))
```

```
Report Classification:
              precision    recall  f1-score   support

           0       0.95      0.72      0.82        25
           1       0.77      0.96      0.86        25

    accuracy                           0.84        50
   macro avg       0.86      0.84      0.84        50
weighted avg       0.86      0.84      0.84        50

Matrix Confusion:
 [[18  7]
 [ 1 24]]
Accuracy:
 0.84
```

نتیجه‌گیری

- با استفاده از مدل‌های pre-trained شده که به دادگان بیشتری دسترسی داشته‌اند می‌توان دقت بهتری کسب نمود، به این منظور نیاز است که آن‌ها را base خود قرار داده و عملیات fine-tune انجام شود تا مدل مطابق آنچه که ما نیاز داریم بشود.

- شبیه این مساله را در مسائل تصویر نیز داریم، برای مثال شما یک شبکه رو برای تشخیص یک موجود استفاده کردی، سپس چندین لایه رو فریز می‌کنی و فقط چند لایه رو برای تشخیص یه موجود دیگر اموزش میدی، البته این کار وسیع تر است و transferred learning نامیده می‌شود. اما کانسپت این کار همان است چرا که کمک می‌کند با داده کمتر شبکه خوبی در نهایت داشته باشی.